

Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application.

Listing of Claims:

1. (Previously Presented) A client-side system stored on a computer, wherein the client-side system logs, in a logging file, a plurality of user interactions performed in an application program module and periodically uploads the logging files to a remote server system for analysis of the logging file, wherein the client-side system comprises:

a logging code in communication with the application program module, wherein the logging code comprises a plurality of hooks into the application program module and an operating system of the computer, wherein when a user performs any recordable action within an application program, one of the plurality of hooks is triggered and a data record is generated;

a logging file in communication with the logging code, wherein the logging code stores the data record in the logging file;

a script file in communication with the logging file, wherein the script file is operative to upload the logging file to the remote server system, wherein uploading the logging file to the remote server system comprises opening an Active Data Object (ADO) session with the remote server system, renaming the logging file with a random number therein preventing duplication of a logging file name at the remote server system and placing the logging file into an ADO database record set; and

a set-up program module, wherein launching the set-up program module comprises installing the logging code in a memory of the computer and setting a registry key in a registry of the operating system as an indicator to the application program to load the logging code when monitoring of the plurality of user interactions has been indicated, and wherein launching the set-up program module signifies user consent to have application program actions logged in exchange for an incentive, the incentive

comprising free software in exchange for participation in a survey to collect user demographic information;

wherein the registry is checked by the application program to determine if the monitoring of the plurality of user interactions has been indicated and, if so, then the monitoring of the plurality of user interactions is started in response to calling an initialization function.

2. (Previously Presented) The system of claim 1 further comprising an event stored in the operating system and created in a predetermined time period, wherein, in response to the event being triggered, the script file uploads the logging file to the remote server system.

3. (Previously Presented) The system of claim 2 wherein the script file uploads the logging file to the remote server system via an Internet connection, wherein the event is created at a random time within the predetermined time period when heavy use of the computer and the Internet connection is less likely than other times, the predetermined time period comprising at least four increments of time.

4. (Original) The system of claim 3 wherein the script file and logging code are generated by a set-up program module included with the application program module and stored on the computer.

5. (Currently Amended) A computer-implemented method for tracking a plurality of user interactions performed in a software application program module stored on the user's computer, the method comprising the steps of:

allowing a user to determine if they wish to have interactions with the software application program module logged in exchange for an incentive, the incentive comprising free software in exchange for participation in a survey to collect user demographic information;

determining if any recordable user interaction performed in the software application program module has occurred by determining whether a notification has been received by a logging code from any one of a plurality of hooks, wherein each of the plurality of hooks causes an event message to be routed to the logging code for an analysis, the analysis comprising an inspection of the event message to determine whether the event message affects a user interface of the application program module prior to the event message being sent to the application program module, wherein the plurality of hooks are implemented by the logging code and wherein, for a particular hook, the logging code uses a plurality of dynamic link libraries to determine a particular window handle that the particular hook points to;

utilizing a best fit algorithm to determine an object and an element that the window handle is associated with, wherein the object comprises a window and the element comprises at least one of command bars, dialogs, and task panes, wherein the logging code collects a plurality of data points that specify a numeric identifier for the element and an identifier on the object that contains the element, wherein the element refers to the event message, and wherein the logging code contains code that filters a plurality of event messages to determine the element referred to by the event message;

recording the user interaction in a logging file on the computer, wherein each recorded user interaction further comprises a screen resolution;

determining that an event is triggered during a predetermined time period;

opening an Active Data Object (ADO) session with [[the]] an remote analysis server;

renaming the logging file to prevent duplication of a logging file name at the remote server system;

placing the logging file into an ADO database record set; and

in response to the event triggering during the predetermined time period, determining whether the logging file exists, and, if so, then uploading the logging file to the [[a]] remote analysis server, wherein uploading the logging file comprises posting the ADO database record set to the remote analysis server.

6. (Previously Presented) The method of claim 5 wherein each recorded user interaction further comprises a time stamp, a user identification, a UI element identifier and a description of the method invoked to interact with the software application program module.

7. (Previously Presented) The method of claim 6 further comprising the step of deleting the logging file on the computer after it has been uploaded.

8. (Previously Presented) The method of claim 7 wherein renaming the logging file comprises renaming the logging file with a random number.

9. (Canceled)

10. (Previously Presented) The method of claim 5 wherein the remote analysis server is a Hypertext Transfer Protocol (HTTP) server.

11. (Previously Presented) A computer-readable hardware storage media comprising computer-executable instructions, which when executed, perform the steps of claim 10.

12–19. (Canceled)

20. (Previously Presented) The method of claim 6 wherein the description of the method invoked to interact with the software application program module comprises at least one of keyboard or mouse.

21. (Previously Presented) A client-side system stored on a computer, wherein the client-side system logs, in a logging file, a plurality of user interactions performed in an application program module and periodically uploads the logging files to a remote server system for analysis of the logging file, wherein the client-side system comprises:

a logging code in communication with the application program module, wherein the logging code comprises a plurality of hooks into the application program module and an operating system of the computer, wherein when a user performs any recordable action within an application program, one of the plurality of hooks is triggered and a data record is generated;

a logging file in communication with the logging code, wherein the logging code stores the data record in the logging file;

a script file in communication with the logging file, wherein the script file is operative to upload the logging file to the remote server system, wherein uploading the logging file to the remote server system comprises opening an Active Data Object (ADO) session with the remote server system, renaming the logging file with a random number therein preventing duplication of a logging file name at the remote server system and placing the logging file into an ADO database record set; and

a set-up program module, wherein launching the set-up program module comprises installing the logging code in a memory of the computer and setting a registry key in a registry of the operating system as an indicator to the application program to load the logging code when monitoring of the plurality of user interactions has been indicated, and wherein launching the set-up program module signifies user consent to have application program actions logged in exchange for an incentive, the incentive comprising free software in exchange for participation in a survey to collect user demographic information;

an event stored in the operating system and created in a predetermined time period, wherein, in response to the event being triggered, the script file uploads the logging file to the remote server system via an Internet connection, wherein the event is

created at a random time within the predetermined time period when heavy use of the computer and the Internet connection is less likely than other times, the predetermined time period comprising at least four increments of time; and

wherein the registry is checked by the application program to determine if the monitoring of the plurality of user interactions has been indicated and, if so, then the monitoring of the plurality of user interactions is started in response to calling an initialization function.